

An $\Omega((n \log n)/R)$ Lower Bound for Fourier Transform Computation in the R -Well Conditioned Model

Nir Ailon
 Department of Computer Science
 Technion Israel Institute of Technology
 Haifa, Israel
 nailon@cs.technion.ac.il

July 25, 2014

Abstract

Obtaining a non-trivial (super-linear) lower bound for computation of the Fourier transform in the linear circuit model has been a long standing open problem for over 40 years.

An early result by Morgenstern from 1973, provides an $\Omega(n \log n)$ lower bound for the unnormalized Fourier transform when the constants used in the computation are bounded. The proof uses a potential function related to a determinant. That result does not explain why the normalized Fourier transform (of unit determinant) should be difficult to compute in the same model. Hence, it is not scale insensitive.

More recently, Ailon (2013) showed that if only unitary 2-by-2 gates are used, and additionally no extra memory is allowed, then the normalized Fourier transform requires $\Omega(n \log n)$ steps. This rather limited result is also sensitive to scaling, but highlights the complexity inherent in the Fourier transform arising from introducing entropy, unlike, say, the identity matrix (which is as complex as the Fourier transform using Morgenstern's arguments, under proper scaling).

In this work we extend the arguments of Ailon (2013). In the first extension, which is also the main contribution, we provide a lower bound for computing any scaling of the Fourier transform. Our restriction is that, the composition of all gates up to any point must be a well conditioned linear transformation. The lower bound is $\Omega(R^{-1}n \log n)$, where R is the uniform condition number. The model is realistic for algorithms computing linear transformations, because low condition number promotes the highly desirable properties of robustness and accuracy.

The second extension assumes extra space is allowed, as long as it contains information of bounded norm at the end of the computation.

The main technical contribution is an extension of matrix entropy used in Ailon (2013) for unitary matrices to a potential function computable for any invertible matrix, using “quasi-entropy” of “quasi-probabilities”.

1 Introduction

The (discrete) normalized Fourier transform is a complex linear mapping sending an input $x \in \mathbb{C}^n$ to $y = Fx \in \mathbb{C}^n$, where F is an $n \times n$ unitary matrix defined by

$$F(k, \ell) = n^{-1/2} e^{-i2\pi(k-1)(\ell-1)/n}.$$

The *unnormalized Fourier transform* matrix is defined as $n^{1/2}F$.¹ The Fast Fourier Transform (FFT) of Cooley and Tukey [3] is a method for computing the Fourier transform (normalized or not - the adjustment is easy) of a vector $x \in \mathbb{C}^n$ in time $O(n \log n)$ using a so called linear algorithm. A linear algorithm, as defined in [5], is a sequence $\mathcal{F}_0, \mathcal{F}_1, \dots$, where each \mathcal{F}_i is a set of affine functions, for each $i \geq 0$ $\mathcal{F}_{i+1} = \mathcal{F}_i \cup \{\lambda_i f + \mu_i g\}$ for some $\lambda_i, \mu_i \in \mathbb{C}$ and $f, g \in \mathcal{F}_i$, and \mathcal{F}_0 contains (projections onto) the input variables as well as constants.

It is trivial that computing the Fourier Transform requires a linear number of steps, but no non-trivial lower bound is known without making very strong assumptions about the computational model. Papadimitriou, for example, computes in [6] an $\Omega(n \log n)$ lower bounds for Fourier transforms in finite fields using a notion of an information flow network. It is not clear how to extend that result to the Complex field. There have also been attempts [8] to reduce the constants hiding in the upper bound of $O(n \log n)$, while also separately counting the number of additions versus the number of multiplications (by constants). In 1973, Morgenstern proved that if the modulus of the λ_i 's and μ_i 's is bounded by 1 then the number of steps required for computing the *unnormalized* Fourier transform in the linear algorithm model is at least $\frac{1}{2}n \log_2 n$. It should be noted that Cooley and Tukey's unnormalized FFT indeed can be expressed as a linear algorithm with coefficients of the form e^{iz} for some real z , namely, complex numbers of unit modulus.

The main idea of Morgenstern is to define a potential function for each \mathcal{F}_i in the linear algorithm sequence, equaling the maximal absolute value of a determinant of a square submatrix in a certain matrix corresponding to \mathcal{F}_i . The technical step is to notice that the potential function can at most double in each step. The determinant of the unnormalized Fourier transform is $n^{n/2}$, hence the lower bound of $\frac{1}{2}n \log_2 n$.

The determinant of the *normalized* Fourier transform, however, is 1. Morgenstern's method can therefore not be used to derive any useful lower bound for computing the normalized Fourier transform in the linear algorithm model with constants of at most unit modulus. Using constants of modulus $1/\sqrt{2}$ in the normalized version of FFT, on the other hand, does compute the normalized Fourier transform in $O(n \log n)$ steps.

The normalized and unnormalized Fourier transforms are proportional to each other, and hence we don't believe there should be a difference between their computational complexities in any reasonable computational model.² It is important to note that, due to the model's weakness, Morgenstern's result teaches us, upon inspection of the proof, that both matrices $\sqrt{n}F$ (the unnormalized Fourier transform) and $\sqrt{n}\text{Id}$ are in the same complexity class. More generally, it tells us that all unitary matrices scaled up by the same constant (\sqrt{n} in this case) are in the same complexity class. Ailon [1] hence studied the complexity of the Fourier transform *within* the unitary group. In his result he showed that, if the algorithm can only apply 2 by 2 unitary transformations at each step, then at least $\Omega(n \log n)$ steps are required for computing the normalized Fourier transform. The proof is done by defining a potential function on the matrices M_i defined by composing the first i gates. The potential function is simply the Shannon entropies of the probability distributions defined by the squared modulus of elements in the matrix rows. (Due to unitarity, each row, in fact, thus defines a probability distribution).

This work takes the idea in [1] a significant step forward, and obtains a $\Omega(n \log n)$ lower bound for *any scaling* of the Fourier transform in a stronger model of computation which we call the *uniformly well conditioned*. At each step, the algorithm can either multiply a variable by a nonzero constant,

¹The unnormalized Fourier transform is sometimes referred to, in literature, as the "Fourier transform". We prefer to call F the Fourier transform, and $\sqrt{n}F$ the unnormalized Fourier transform.

²It should also be noted that the determinant of any submatrix of the Fourier matrix has determinant at most 1.

or perform a unitary transformation involving 2 variables. The matrix M_i defining the composition of the first i steps must be well conditioned with constant R . This means that $\|M_i\| \cdot \|M_i^{-1}\| \leq R$, where $\|\cdot\|$ is spectral norm. Taking this number into account, the actual lower bound we obtain is $\Omega(R^{-1}n \log n)$. This main result is presented in Section 3. It should be noted that well conditionedness is related to numerical stability: The less well conditioned a transformation is, the larger the set of inputs on which numerical errors would be introduced in any computational model with limited precision. An important commonly studied example is the linear regression (least squares) problem, in which the condition number controls a tradeoff between computational complexity and precision [4]. We also note the work of Raz et. al [7], in which a notion of numerical stability was also used to lower bound the complexity of certain functions, although that work does not seem to be directly comparable to this.

Another limitation of [1] is that no additional memory (extra variables) were allowed in the computation. (This limitation is not present in [5].) In Section 5 this limitation is removed, assuming a bound on the amount of information held in the extra space at the end of the computation.

1.1 Different Types of Fourier Transforms

In this work we will assume that n is even and will use F to denote one of the following:

1. The real orthogonal $n \times n$ matrix computing the (normalized) complex discrete Fourier transform (DFT) of order $n/2$ on an input $\hat{x} \in \mathbb{C}^{n/2}$, where the real part of \hat{x} is stored in $n/2$ coordinates and the imaginary part in the remaining $n/2$.
2. The (normalized) Walsh-Hadamard Fourier transform, where n is assumed to be an integer power of two and $F(i, j) = \frac{1}{\sqrt{n}}(-1)^{\langle i-1, j-1 \rangle}$, where for $a \in [0, 2^{\log n} - 1]$, $[a]$ is the binary vector representing a in base 2, and $\langle \cdot, \cdot \rangle$ is dot-product over Z_2 . It is well known that Fx given $x \in \mathbb{R}^n$ can be computed in $O(n \log n)$ operations using the so called Walsh-Hadamard transform. All the above discussion on Morgenstern's result applies to this transformation as well.

In fact, the field of harmonic analysis defines a Fourier transforms corresponding to any Abelian group of order n , but this abstraction would not contribute much to the discussion. Additionally, our results apply to the well known (and useful) cosine transform, which is a simple derivation of DFT. In any case, DFT and Walsh-Hadamard are central to engineering and the reader is invited to concentrate on those two.

1.2 A Note on Quantum Computation

It is important to note that we are in the classical setting, not quantum. A quantum version of the Fourier transform can be computed in time $O(\log^2 n)$ using an algorithm by Shor (refer e.g. to Chapter 5, [2]) but that setting is different.

1.3 A Note on Universality of our Model

We argue that the model of computation studied in this work is suitable for studying any algorithm which computes a linear transformation by performing a sequence of simple linear operations. Imagine a machine which can perform linear operations acting on k variables in one step, for some constant k . Recall the SVD theorem stating that any such mapping ψ can be written as a

composition of three linear mappings, where two are orthogonal and one is diagonal (multiplication by constants). Additionally, if ψ is nonsingular, then all the diagonal constants are nonzero. Also recall that any orthogonal mapping of rank k can be decomposed into $O(k^2)$ orthogonal mappings, each acting on at most two coordinates. Hence, up to a constant speedup factor, such a machine can be efficiently simulated using our model.

Additionally, any fixed precision machine cannot afford extreme ill conditionedness. The worse the condition number of the computation is (at some point), the higher the maximal ratio between two numbers the machine would have to be able to represent. Otherwise viewed, the worse the condition number, the smaller the set of inputs for which the computation is guaranteed to be numerically stable.

Hence, this work in fact offers the first tradeoff between running time and precision of scale-free Fourier transform algorithms in a computational model that is relevant to any fixed precision machine that performs simple linear operations as atomic steps. Whether this tradeoff is tight is subject to further investigation.

1.4 Our Main Technique: Quasi-Entropy as a Potential Function

Ailon [1] defined the entropy of a unitary matrix $M \in \mathbb{C}^{n \times n}$ to be

$$\Phi(M) = \sum_{i=1}^n \sum_{j=1}^n f(M(i, j)) , \quad (1.1)$$

where for any nonnegative x ,

$$f(x) = \begin{cases} 0 & x = 0 \\ -|x|^2 \log |x|^2 & x > 0 \end{cases} . \quad (1.2)$$

Since M is unitary, for any row i the numbers $(|M(i, 1)|^2, \dots, |M(i, n)|^2)$ form a probability distribution vector, from which we can view $\Phi(M)$ as the sum of the Shannon entropy of n distributions. Note that $\Phi(M)$ is always in the range $[0, n \log n]$. (Throughout, we will take all logarithms to be in base 2, as common in information theory). Ailon [1] claimed, using a simple norm preservation argument, that for any (complex) Givens matrix S ,

$$|\Phi(M) - \Phi(SM)| \leq 2 , \quad (1.3)$$

where we remind the reader that a Givens matrix is any unitary transformation acting on two coordinates. Since $\Phi(\text{Id}) = 0$ and $\Phi(F) = n \log n$, the conclusion was that at least $\frac{1}{2}n \log n$ Givens operations are required to compute the (normalized) Fourier transformation F .

The starting point of this work is extending the definition of Φ in (1.1) to any (nonsingular) matrix. Indeed, there is no reason to believe that an optimal Fourier transform algorithm must be confined to the unitary group. Using (1.1) verbatim does not help proving a lower bound, as one can easily see that $\Phi(M)$ can change by $\Omega(\log n)$ if we multiply a row of M by a nonzero constant C such that $|C| \neq 1$. (For example, if a row of M equals $(1/\sqrt{n}, \dots, 1/\sqrt{n})$, then by multiplying the row by $C = 2$ additively changes the entropy by $\Omega(\log n)$.)

We now fix this problem. For simplicity, we will work over \mathbb{R} and not over \mathbb{C} . The complex Fourier transform can be simulated over \mathbb{R} by doubling the dimension.³ (Note that over \mathbb{R} unitary

³ This can be done by representing the input (and output) using $2n$ variables, half dedicated to the real part and half to the imaginary part of the complex input. Accordingly, each matrix element $F(k, \ell) = n^{-1/2} e^{-i2\pi k\ell/n}$ of the complex Fourier transform becomes a 2×2 rotation matrix with angle $-2\pi k\ell/n$, multiplied by $n^{-1/2}$.

matrices are referred to as orthogonal matrices, and we shall follow this convention.) For any real nonsingular matrix M , we define

$$\Phi(M) := - \sum_{i=1}^n \sum_{j=1}^n \hat{f}(M(i, j), M^{-1}(j, i)) , \quad (1.4)$$

where for all $x, y \in \mathbb{R}$,

$$\hat{f}(x, y) := \begin{cases} 0 & x \cdot y = 0 \\ -x \cdot y \cdot \log |x \cdot y| & x \cdot y \neq 0 \end{cases} . \quad (1.5)$$

Note that if M is orthogonal then $M(i, j) = M^{-1}(j, i)$. This implies that M defined in (1.4) is an extension of (1.1) from the unitary to the nonsingular group. Also note that for all i , the numbers $M(i, 1)M^{-1}(1, i), \dots, M(i, n)M^{-1}(n, i)$ sum up to one (by definition of matrix inversion) but they do not form a probability distribution vector because they may be negative or > 1 in general, hence we think of them as quasi-probabilities (and of Φ as quasi-entropy). Our main Lemma 3.2 below shows that a Givens rotation applied to M can change $\Phi(M)$ by at most $O(R)$, where R is the condition number of M .

1.5 Contribution and Limitations

We believe that our main contribution is in showing a lower bound on computation of the Fourier transform that does not depend on scaling. Indeed, if M_i is the matrix defined by the composition of the first i steps (see exact model definition below), then $\Phi(M_i)$ is completely insensitive to rescaling (of all rows) by any arbitrary large or small nonzero number, because such an operation has no effect on neither Φ nor on condition numbers. We argue that the generalized matrix entropy Φ defined in (1.4), which is interesting in its own right, is an important key to understanding the complexity of one of the most important linear transformations used in science and engineering, for which an algorithm that is believed to be optimal has been around for half a decade. We point out the following shortcoming of this work, which also gives rise to interesting open problems: Although constant well conditionedness enhances numerical stability, for many applications it is reasonable to work with condition numbers that grow with n , even polynomially. Our result implies nontrivial bounds for condition number up to $o(\log n)$. It would be interesting to prove interesting lower bounds for less well conditioned computations.

2 The Well Conditioned Model of Computation

For a matrix M , we let $M^{(i)}$ denote the i 'th column of M . Our model of computation consists of layers L_0, \dots, L_m , each containing exactly n nodes and representing a vector in \mathbb{R}^n . The first layer, $L_0 \in \mathbb{R}^n$, is the input. The last layer $L_m \in \mathbb{R}^n$ is the output.

For $i = 1, \dots, m$, the i 'th gate connects layer $i - 1$ with later i . There are two types of gates: *rotations* and *constants*. If gate i is a rotation, then there are two indices $k_i, \ell_i \in [n]$, $k_i < \ell_i$, and an orthogonal matrix

$$A_i = \begin{pmatrix} a_i(1, 1) & a_i(1, 2) \\ a_i(2, 1) & a_i(2, 2) \end{pmatrix} = \begin{pmatrix} \cos \theta_i & \sin \theta_i \\ -\sin \theta_i & \cos \theta_i \end{pmatrix} .$$

For each $j \notin \{k_i, \ell_i\}$, $L_i(j) = L_{i-1}(j)$. The values of $L_i(k_i)$ and $L_i(\ell_i)$ are given as

$$\begin{pmatrix} L_i(k_i) \\ L_i(\ell_i) \end{pmatrix} = A_i \begin{pmatrix} L_{i-1}(k_i) \\ L_{i-1}(\ell_i) \end{pmatrix} .$$

Note that the transformation taking L_{i-1} to L_i is known as a *Givens rotation*.

If gate i is of type constant, then it is defined by an index $k_i \in [n]$ and a nonzero c_i . For each $j \neq k_i$, $L_i(j) = L_{i-1}(j)$. Additionally, $L_i(k_i) = c_i L_{i-1}(k_i)$.

We will encode the circuit using the sequence

$$(k_i, \ell_i, \theta_i, c_i)_{i=1}^m ,$$

where we formally define c_i to be 0 for rotation gates, and $\ell_i = 0$ for constant gates.

Let M_i be the matrix transforming L_0 (as a column vector) to L_i . We say that M_i is the i 'th defining matrix of the circuit. If gate i is a rotation, then M_i is obtained from M_{i-1} by replacing rows k_i and ℓ_i in M_{i-1} by the application of A_i to these rows, stacked one on top of the other to the right of A_i . If gate i is diagonal, then M_i is obtained from M_{i-1} by multiplying row k_i of M_{i-1} by c_i . Also, $M_0 = \text{Id}$.

Definition 2.1. A layered circuit of depth m is R -uniformly well conditioned (for some $R > 1$) if

$$\max_{i \in [m]} \{ \|M_i\| \cdot \|M_i^{-1}\| \} \leq R .$$

Note that a 1-uniformly well conditioned circuit recovers the model of [1] (restricted over the reals).

3 The Main Result

Theorem 3.1. *If an R -uniformly well conditioned layered circuit $\mathcal{C} = (k_i, \ell_i, \theta_i, c_i)_{i=1}^m$ computes a transformation that is proportional to the Fourier transform F , then the number of rotations is $\Omega(R^{-1}n \log n)$.*

Proof. We begin with an observation, which can be proven with a simple induction: For any $i \in [m]$, $(M_i^{-1})^T$ is the i 'th defining matrix of a circuit \mathcal{C}' defined by $(k_i, \ell_i, \theta_i, c'_i)_{i=1}^m$, where $c'_i = 1/c_i$ if the i 'th gate of \mathcal{C} is of type constant, and 0 otherwise. A clear consequence of this observation is that if the i 'th gate of \mathcal{C} is of type constant, then

$$\Phi(M_{i-1}) = \Phi(M_i) .$$

Indeed, just notice that for $p = k_i$ and any $q \in [n]$, $M_i(p, q) = c_i M_{i-1}(p, q)$ and $(M_i^{-1})^T(p, q) = c_i^{-1} (M_{i-1}^{-1})^T(p, q)$. We analyze the effect of rotation gates on Φ . To this end, we need the following lemma.

Lemma 3.2. *Recall \hat{f} as in (1.5). For 4 real numbers w, x, y, z , define*

$$\Psi(w, x, y, z) = \hat{f}(w, x) + \hat{f}(y, z) .$$

Now define

$$\begin{aligned}\alpha(w, x, y, z) &= \sup_{\theta \in [0, 2\pi]} \Psi(w \cos \theta + y \sin \theta, x \cos \theta + z \sin \theta, -w \sin \theta + y \cos \theta, -x \sin \theta + z \cos \theta) \\ \beta(w, x, y, z) &= \inf_{\theta \in [0, 2\pi]} \Psi(w \cos \theta + y \sin \theta, x \cos \theta + z \sin \theta, -w \sin \theta + y \cos \theta, -x \sin \theta + z \cos \theta) .\end{aligned}$$

Then

$$\sup_{w, x, y, z} \frac{\alpha(w, x, y, z) - \beta(w, x, y, z)}{\sqrt{(w^2 + y^2)(x^2 + z^2)}} = O(1) . \quad (3.1)$$

(we formally define the last fraction as 0 if either $w^2 + y^2 = 0$ or $x^2 + z^2 = 0$. Note that in this degenerate case both $\alpha(w, x, y, z) = 0$ and $\beta(w, x, y, z) = 0$).

The proof of the lemma is deferred to Section 4. Now let i be such that the i 'th gate is a rotation. Then using the definition of Ψ as in the lemma,

$$\begin{aligned}\Phi(M_i) - \Phi(M_{i-1}) &= \sum_{q=1}^n [\Psi(M_i(k_i, q), M_i^{-1}(q, k_i), M_i(\ell_i, q), M_i^{-1}(q, \ell_i)) \\ &\quad - \Psi(M_{i-1}(k_i, q), M_{i-1}^{-1}(q, k_i), M_{i-1}(\ell_i, q), M_{i-1}^{-1}(q, \ell_i))] \end{aligned}$$

By Lemma 3.2, hence for some global $C > 0$

$$\begin{aligned}|\Phi(M_i) - \Phi(M_{i-1})| &\leq C \sum_{q=1}^n \sqrt{(M_i(k_i, q)^2 + M_i(\ell_i, q)^2) (M_i^{-1}(q, k_i)^2 + M_i^{-1}(q, \ell_i)^2)} \\ &\leq C \sqrt{\left(\sum_{q=1}^n M_i(k_i, q)^2 + M_i(\ell_i, q)^2 \right) \left(\sum_{q=1}^n M_i^{-1}(q, k_i)^2 + M_i^{-1}(q, \ell_i)^2 \right)} \\ &\leq 2C \|M_i\| \cdot \|M_i^{-1}\| \leq 2CR ,\end{aligned} \quad (3.2)$$

where the second inequality is Cauchy-Schwarz, and the third is from the definition of condition number (together with the observation that the norm of any row or column of a matrix is at most the spectral norm of the matrix). Hence,

$$|\Phi(M_i) - \Phi(M_{i-1})| \leq O(R) . \quad (3.3)$$

Now notice that $\Phi(M_0) = \Phi(\text{Id}) = 0$ and $\Phi(M_m) = \Phi(F) = n \log n$. Hence $m = \Omega(R^{-2} n \log n)$, as required. \square

4 Proof of Lemma 3.2

If either $(w, y) = (0, 0)$ or $(x, z) = (0, 0)$ then the LHS of (3.1) is clearly 0. Assume first that the vectors (w, y) and (x, z) are not proportional to each other. Without loss of generality, we can

assume that the vector direction $(1, 0) \in \mathbb{R}^2$ is an angle bisector of the two segments connecting the origin with (w, y) and (x, z) . In words, there exist numbers $r, s > 0$ and an angle ϕ such that

$$\begin{aligned}(w, y) &= \left(r \cos \frac{\phi}{2}, r \sin \frac{\phi}{2} \right) \\ (x, z) &= \left(s \cos \frac{\phi}{2}, -s \sin \frac{\phi}{2} \right) .\end{aligned}$$

By symmetry, we can assume that $\phi \in [-\pi/2, \pi/2] \setminus \{0\}$, because otherwise we could replace w with $-w$ and y with $-y$, which would result in negation of Ψ (leaving $(\alpha - \beta)$ untouched). In fact, we can assume that $\phi \in (0, \pi/2]$, because otherwise we would replace the roles of (w, y) and (x, z) . With this notation, we have for all $\theta \in [0, 2\pi)$

$$w \cos \theta + y \sin \theta = r \cos \left(\frac{\phi}{2} + \theta \right) \quad x \cos \theta + z \sin \theta = s \cos \left(-\frac{\phi}{2} + \theta \right) \quad (4.1)$$

$$-w \sin \theta + y \cos \theta = r \sin \left(\frac{\phi}{2} + \theta \right) \quad -x \sin \theta + z \cos \theta = s \sin \left(-\frac{\phi}{2} + \theta \right) \quad (4.2)$$

Therefore,

$$\begin{aligned}\Psi(w \cos \theta + y \sin \theta, x \cos \theta + z \sin \theta, -w \sin \theta + y \cos \theta, -x \sin \theta + z \cos \theta) = \\ -rs \cos \left(\frac{\phi}{2} + \theta \right) \cos \left(-\frac{\phi}{2} + \theta \right) \log \left| rs \cos \left(\frac{\phi}{2} + \theta \right) \cos \left(-\frac{\phi}{2} + \theta \right) \right| \quad (4.3)\end{aligned}$$

$$-rs \sin \left(\frac{\phi}{2} + \theta \right) \sin \left(-\frac{\phi}{2} + \theta \right) \log \left| rs \sin \left(\frac{\phi}{2} + \theta \right) \sin \left(-\frac{\phi}{2} + \theta \right) \right|. \quad (4.4)$$

We view the last expression as a function of θ , and write $\Psi(\theta)$ for shorthand. The function Ψ is differentiable everywhere except $\theta \in Q = \{\pm \frac{\phi}{2} + j\frac{\pi}{2}\}$ for $j = 0, 1, 2, \dots$. For $\theta \in Q$, it is not hard to see that Ψ is not a local optimum. It hence suffices to find local optima of Ψ for $\theta \notin Q$. Consider first the range $\theta \in \left(-\frac{\phi}{2}, \frac{\phi}{2}\right)$. In this range, the argument inside the absolute value in (4.3) is positive, while the one inside (4.4) is negative. Differentiating with respect to θ , we get

$$\begin{aligned}\frac{d}{d\theta} \Psi(\theta) &= rs \left[\sin \left(\frac{\phi}{2} + \theta \right) \cos \left(-\frac{\phi}{2} + \theta \right) + \cos \left(\frac{\phi}{2} + \theta \right) \sin \left(-\frac{\phi}{2} + \theta \right) \right] \\ &\quad \times \left[1 + \log \left(rs \cos \left(\frac{\phi}{2} + \theta \right) \cos \left(-\frac{\phi}{2} + \theta \right) \right) \right] \\ &+ rs \left[\cos \left(\frac{\phi}{2} + \theta \right) \sin \left(-\frac{\phi}{2} + \theta \right) + \sin \left(\frac{\phi}{2} + \theta \right) \cos \left(-\frac{\phi}{2} + \theta \right) \right] \\ &\quad \times \left[1 - \log \left(-rs \sin \left(\frac{\phi}{2} + \theta \right) \sin \left(-\frac{\phi}{2} + \theta \right) \right) \right] \\ &= rs(\sin 2\theta) \left(2 + \log \left(-\frac{\cos \left(\frac{\phi}{2} + \theta \right) \cos \left(-\frac{\phi}{2} + \theta \right)}{\sin \left(\frac{\phi}{2} + \theta \right) \sin \left(-\frac{\phi}{2} + \theta \right)} \right) \right) .\end{aligned}$$

One checks using standard trigonometry that the last log is nonnegative. Hence, $d\Psi/d\theta$ vanishes only when $\theta = 0$. For this value,

$$\Psi(0) = -rs \cos^2\left(\frac{\phi}{2}\right) \log\left(rs \cos^2\left(\frac{\phi}{2}\right)\right) + rs \sin^2\left(\frac{\phi}{2}\right) \log\left(rs \sin^2\left(\frac{\phi}{2}\right)\right). \quad (4.5)$$

We now study the case $\theta \in (\phi/2, \pi/2 - \phi/2)$. In this range, the argument inside the absolute values in both (4.3) and (4.4) is positive. For this case, using a similar derivation as above, the derivative $d\Psi/d\theta$ equals

$$\frac{d}{d\theta}\Psi(\theta) = rs(\sin 2\theta) \log\left(\frac{\cos\left(\frac{\phi}{2} + \theta\right) \cos\left(-\frac{\phi}{2} + \theta\right)}{\sin\left(\frac{\phi}{2} + \theta\right) \sin\left(-\frac{\phi}{2} + \theta\right)}\right).$$

By our assumption on ϕ , the last derivation vanishes eg when $\theta = \pi/4$. For this value,

$$\begin{aligned} \Psi(\pi/4) &= -rs \cos\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos\left(-\frac{\phi}{2} + \frac{\pi}{4}\right) \log\left(rs \cos\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos\left(-\frac{\phi}{2} + \frac{\pi}{4}\right)\right) \\ &\quad - rs \sin\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \sin\left(-\frac{\phi}{2} + \frac{\pi}{4}\right) \log\left(rs \sin\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \sin\left(-\frac{\phi}{2} + \frac{\pi}{4}\right)\right) \end{aligned}$$

By basic trigonometry, one verifies that

$$\begin{aligned} \cos\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos\left(-\frac{\phi}{2} + \frac{\pi}{4}\right) &= \frac{1}{2} \cos^2\left(\frac{\phi}{2}\right) - \frac{1}{2} \sin^2\left(\frac{\phi}{2}\right) = \frac{1}{2} \cos \phi = \\ &\quad \sin\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \sin\left(-\frac{\phi}{2} + \frac{\pi}{4}\right) \end{aligned}$$

Plugging in our derivation of $\Psi(\pi/4)$, we get

$$\Psi(\pi/4) = -rs \left(\cos^2\left(\frac{\phi}{2}\right) - \sin^2\left(\frac{\phi}{2}\right) \right) \log\left(rs \cos\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos\left(-\frac{\phi}{2} + \frac{\pi}{4}\right)\right). \quad (4.6)$$

It is not hard to verify that $\Psi(0)$ and $\Psi(\pi/4)$ are the only extremal values of Ψ . Now notice that in the expression $|\Psi(\pi/4) - \Psi(0)|$, the term $\log(rs)$ is cancelled out, and we are left with $|\Psi(\pi/4) - \Psi(0)| = rsg(\phi)$, where

$$\begin{aligned} g(\phi) &= \left| \cos^2\left(\frac{\phi}{2}\right) \log\left(\frac{\cos^2\left(\frac{\phi}{2}\right)}{\cos\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos\left(-\frac{\phi}{2} + \frac{\pi}{4}\right)}\right) + \sin^2\left(\frac{\phi}{2}\right) \log\left(\frac{\cos\left(\frac{\phi}{2} + \frac{\pi}{4}\right) \cos\left(-\frac{\phi}{2} + \frac{\pi}{4}\right)}{\sin^2\left(\frac{\phi}{2}\right)}\right) \right| \\ &= \left| \cos^2\left(\frac{\phi}{2}\right) \log \cos^2\left(\frac{\phi}{2}\right) - \sin^2\left(\frac{\phi}{2}\right) \log \sin^2\left(\frac{\phi}{2}\right) - \cos \phi \log \frac{\cos \phi}{2} \right|. \end{aligned} \quad (4.7)$$

The function $g(\phi)$ is bounded in the range $\phi \in (0, \pi/4]$, by which we conclude that for some global constant C ,

$$\left| \sup_{\theta} \Psi(\theta) - \inf_{\theta} \Psi(\theta) \right| \leq C rs.$$

This concludes the proof for the case $\phi \notin \{0, \pi\}$ (modulo 2π). If (w, y) and (x, z) are proportional to each other ($\phi \in \{0, \pi\}$), then the analysis uses the same simple norm preservation argument as in [1]. (Details omitted)

5 Using Additional Space

We assume in this section that aside from the n input variables, the algorithm has access to an additional memory of total size N . We would like to explore to what extent this additional memory could help in Fourier computation, using the framework developed in the previous sections. We will assume throughout that

$$N \leq n \log n , \quad (5.1)$$

because, assuming all extra memory is accessed in the linear circuit, $N/2$ is a lower bound on the depth of the circuit. Additionally, we will assume that this additional memory is initialized as 0. This is not a real restriction, because the Fourier transform is a homogenous transformation.⁴ For convenience, we will work with linear circuits as defined in Section 2 over \mathbb{R}^{n+N} , and

As a warmup, we will also assume that the N extra output variables are identically 0. In other words, that there is no “garbage information” in the N output variables. This means that, if the circuit has depth m then $[M_m]_{[n],[n]} = F$ and $[M_m]_{[n+N] \setminus [n],[n]} = 0$, where for a matrix A and integer sets I, J , $[A]_{I,J}$ denotes the submatrix of A corresponding to rows I and columns J . We will later relax this assumption.

Theorem 5.1. *If an R -uniformly well conditioned layered circuit \mathcal{C} computes a transformation M such that $M_{[n],[n]} = F$ and $M_{[n+N] \setminus [n],[n]} = 0$, then the number of rotations in the circuit is $\Omega(R^{-1}n \log n)$.*

Proof. We proceed as in the proof of Theorem 3.1, except we now work with a partial entropy function defined as follows:

$$\Phi_n(M) := - \sum_{i=1}^{n+N} \sum_{j=1}^n \hat{f}(M(i,j), M^{-1}(j,i)) , \quad (5.2)$$

It is easy to see that, as before, for any i such that the i 'th gate is a rotation, $|\Phi_n(M_i) - \Phi_n(M_{i+1})| = O(R)$. We also notice that, by the assumptions, we must have $[M_m^{-1}]_{[n],[n]} = F^{-1}$ and $[M_m^{-1}]_{[n],[N+n] \setminus [n]} = 0$. This implies, as before, that $\Phi_m(M_0) = 0$ and $\Phi_m(M_m) = \Omega(n \log n)$, leading to the claimed result. \square

It is arguably quite restrictive to assume that the extra space must be clean of any “garbage” information at the end of the computation. In particular, by inspection of the last proof, the “garbage” could have a negative contribution to Φ_n , possibly reducing the computational lower bound. This assumption is relaxed in what follows. We will first need a technical lemma.

Lemma 5.2. *There exists a global constant $C_0 < 1/4$ such that the following holds for all n . Let $\varepsilon \in \mathbb{R}^n$ be such that $\|\varepsilon\|_2 \leq C_0$. Then*

$$- \sum_{i=1}^n \frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{n}} + \varepsilon_i \right) \log \left| \frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{n}} + \varepsilon_i \right) \right| \geq \frac{3}{4} \log n . \quad (5.3)$$

The proof of the lemma is deferred to Appendix B. We are now ready to state and prove the main result in the section. We will prove the result for the Walsh-Hadamard Fourier transform for simplicity only, although a technical extension of the last lemma can be used to prove a similar result for any Fourier transform.

⁴More precisely, if require to initialize a subset of the additional memory with values $\neq 0$, then its total linear contribute to the output variables must be 0.

Theorem 5.3. *Assume F is the Walsh-Hadamard matrix. Let \mathcal{C} be an R -uniformly well conditioned layered circuit of depth m . Assume that $[M_m]_{[n],[n]} = F$ and that additionally the spectral norm of $[M_m]_{[N+n]\setminus[n],[n]}$ is at most C_0/R , where C_0 is from Lemma 5.2. Then $m = \Omega(R^{-1}n \log n)$.*

Note that in both Theorems 5.1 and 5.3 the normalization chosen in the theorems is immaterial. For example, we could have replaced F and C_0/R in Theorem 5.3 with $c \cdot F$ and $c \cdot C_0/R$, respectively, for any nonzero c . Hence these lower bounds are insensitive to scaling. We chose the specific normalization to eliminate extra constants in the analysis.

Proof. We will work with the potential function Φ_n defined in (5.2), except that we cannot know the exact value of $\Phi_n(M_m)$ as in the proof of Theorem 5.1. Denote the columns of $[M_m]_{[N+n]\setminus[n],[n]}$ by $u_1, \dots, u_n \in \mathbb{R}^N$, and the columns of $[(M_m^{-1})^T]_{[N+n]\setminus[n],[n]}$ by $v_1, \dots, v_n \in \mathbb{R}^N$. By the bound on the spectral norm of $[M_m]_{[N+n]\setminus[n],[n]}$, we have in particular a uniform bound on its column norms:

$$\max\{\|u_1\|, \dots, \|u_n\|\} \leq 1/(4R). \quad (5.4)$$

Hence the norm of any of the first n columns of M_n is in the range $[1, 1 + 1/(4R)]$. This implies that the spectral norm $\|M_m\|$ is at least 1. By the well conditionedness of M_m , we conclude that $\|M_m^{-1}\|$ is at most R , by which we conclude that

$$\max\{\|v_1\|, \dots, \|v_n\|\} \leq R. \quad (5.5)$$

Using Lemma A.1 in Appendix A together with the constraints (5.4) and (5.5), we have that for any $j \in [n]$,

$$\sum_{i=1}^N \hat{f}(u_j(i), v_j(i)) \geq -\frac{1}{4} \log 4 - \frac{1}{4} \log N = -\frac{1}{2} - \frac{1}{4} \log N \geq -\frac{1}{2} - \frac{1}{2} \log n, \quad (5.6)$$

where in the rightmost inequality we used the assumption that $N \leq n \log n \leq n^2$.

We now need to lower bound the contribution of the upper left square of M_m to the total entropy, namely $\sum_{i,j=1}^n \hat{f}([M_m](i,j), [M_m]^{-1}(j,i))$. By definition if matrix inverse,

$$[M_m]_{[n],[n]}^{-1} F + [M_m]_{[n],[N+n]\setminus[n]}^{-1} [M_m]_{[N+n]\setminus[n],[n]} = \text{Id}_n.$$

Hence,

$$[M_m]_{[n],[n]}^{-1} = F - [M_m]_{[n],[N+n]\setminus[n]}^{-1} [M_m]_{[N+n]\setminus[n],[n]} F.$$

Letting \mathcal{E} denote the error term $-[M_m]_{[n],[N+n]\setminus[n]}^{-1} [M_m]_{[N+n]\setminus[n],[n]} F$, we can succinctly write

$$[M_m]_{[n],[n]}^{-1} = F + \mathcal{E}$$

and then use the norm chain rule to bound:

$$\|\mathcal{E}\| \leq \|[M_m]_{[n],[N+n]\setminus[n]}^{-1}\| \cdot \|[M_m]_{[N+n]\setminus[n],[n]}\| \cdot \|F\| \quad (5.7)$$

$$\leq \|[M_m]_{[n],[N+n]\setminus[n]}^{-1}\| \cdot (C_0/R) \cdot 1, \quad (5.8)$$

where we used the spectral norm bound assumption from the theorem statement. To bound $\|[M_m]_{[n],[N+n]\setminus[n]}^{-1}\|$, note that trivially $\|[M_m]_{[n],[N+n]\setminus[n]}^{-1}\| \leq \|[M_m]^{-1}\|$, and $\|[M_m]^{-1}\|$ is bounded by R (because $\|[M_m]\| \geq \|F\| = 1$ and $[M_m]$ is R -well conditioned). Hence,

$$\|\mathcal{E}\| \leq C_0. \quad (5.9)$$

The last inequality also implies that any column of $[M_n]_{[n],[n]}^{-1}$ has norm at most C_0 . Using Lemma 5.2 for each $i \in [n]$, we get

$$\sum_{j=1}^n \hat{f}([M_m]^{-1}(j, i), [M_m](i, j)) \geq \frac{1}{4} \log n . \quad (5.10)$$

(Note that to be precise, to use Lemma 5.2 we need to flip the sign of $\mathcal{E}(j, i)$ whenever $F(i, j)$ is negative, but this is a small technicality.) Combining (5.6) and (5.10) we conclude that

$$\Phi_n(M_m) \geq \frac{1}{4} n \log n - \frac{1}{2} n .$$

Since $\Phi_n(M_0) = 0$ and $|\Phi_n(M_t) - \Phi_n(M_{t-1})| = O(R)$ for all $t > 1$, we conclude that $m = \Omega(R^{-1} n \log n)$ as required. \square

5.1 Future Work

We raise the following three questions:

1. Is the dependence in R in Theorem 3.1 tight? Intuitively we did not take full advantage of the well conditionedness property. Indeed, we only used it to bound the product of the norm of a row in a matrix and the corresponding row in its inverse.
2. Is the dependence of the lower bound on the norm of columns of the “additional space at output” in R in Theorem 5.3 tight?
3. In Section 4.5.4 in [2], it is shown that most (with respect to the H aar measure) unitary operators require $\Omega(n^2)$ steps using 2×2 unitary operations. This is also true if we relax unitarity and allow well-conditioned circuits. This implies that the techniques developed here cannot be used to prove tight lower bounds for most unitary matrices, because the potential of unitary matrices is globally upper bounded by $O(n \log n)$. Nevertheless, we ask: Can $\Phi(M)$ defined in this work be used for proving lower bounds for other interesting linear algebraic operations?

References

- [1] Nir Ailon. A lower bound for fourier transform computation in a linear model over 2×2 unitary gates using matrix entropy. *Chicago J. of Theo. Comp. Sci.*, 2013.
- [2] Isaac L. Chuang and Michael A. Nielsen. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [3] J. W Cooley and J. W Tukey. An algorithm for the machine computation of complex Fourier series. *J. of American Math. Soc.*, pages 297–301, 1964.
- [4] Gene H. Golun and Charles F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 2 edition, 1989.
- [5] Jacques Morgenstern. Note on a lower bound on the linear complexity of the fast Fourier transform. *J. ACM*, 20(2):305–306, April 1973.

- [6] Christos H. Papadimitriou. Optimality of the fast Fourier transform. *J. ACM*, 26(1):95–102, January 1979.
- [7] Ran Raz and Amir Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. *J. Comput. Syst. Sci.*, 77(1):167–190, 2011.
- [8] S. Winograd. On computing the discrete Fourier transform. *Proc. Nat. Assoc. Sci.*, 73(4):1005–1006, 1976.

A A Useful Quasi-Entropy Bound

Lemma A.1. *Let $x \in \mathbb{R}^n, y \in \mathbb{R}^n$ be two vectors such that $\|x\|_2 = \alpha, \|y\|_2 = \beta$. Then*

$$-\alpha\beta \log n - |\alpha\beta \log \alpha\beta| \leq \sum_{i=1}^n \hat{f}(x_i, y_i) \leq -\alpha\beta \log n + |\alpha\beta \log \alpha\beta|. \quad (\text{A.1})$$

Proof. Notice that

$$\sum_{i=1}^n \hat{f}(x_i, y_i) = \alpha\beta \sum_{i=1}^n \hat{f}(x_i/\alpha, y_i/\beta) - \sum_{i=1}^n x_i y_i \log |\alpha\beta|.$$

Since $|\sum x_i y_i| \leq \alpha\beta$ by Cauchy-Schwartz, it hence suffices to prove the lemma for the case $\alpha = \beta = 1$, which we assume from now on. Let $F(x, y) = \sum_{i=1}^n \hat{f}(x_i, y_i)$. Then

$$\nabla F(x, y) = (-y_1(\log |x_1 y_1| + 1) \cdots -y_n(\log |x_1 y_1| + 1), -x_1(\log |x_1 y_1| + 1) \cdots -x_n(\log |x_n y_n| + 1)).$$

The gradients of the constraints $G_1(x, y) = \|x\|^2$ and $G_2(x, y) = \|y\|^2$ are:

$$\begin{aligned} \nabla G_1(x, y) &= (2x_1 \dots 2x_n, 0 \dots 0) \\ \nabla G_2(x, y) &= (0 \dots 0, 2y_1 \dots 2y_n). \end{aligned}$$

□

Using standard optimization principles, any optima (x, y) of F under $G_1 = G_2 = 1$ satisfies that there exists λ_1, λ_2 such that $\nabla F(x, y) = \lambda_1 \nabla G_1(x, y) + \lambda_2 \nabla G_2(x, y)$. This implies that for all $i \in [n]$, $x_i/y_i = \lambda_1(\log |x_i y_i| + 1)$ and $y_i/x_i = \lambda_2(\log |x_i y_i| + 1)$, hence $x_i^2/y_i^2 = \lambda_1/\lambda_2$. But we assumed that $\|x\|^2 = \|y\|^2 = 1$, hence $\lambda_1 = \lambda_2$ and for all i either $x_i = y_i$ or $x_i = -y_i$.

It remains to find the optima of $\tilde{F}(x, b) = \sum \hat{f}(x_i, b_i x_i)$ under the constraints $\|x\|_2^2 = 1$ and $b_i = \pm 1$ for all i . It is easy to see that for any fixed $b = b_0$,

$$\sup_x \tilde{F}(x, b_0) \leq \max\{\sup_x \tilde{F}(x, (1 \dots 1)), \sup_x \tilde{F}(x, -(1 \dots 1))\}.$$

Hence it suffices to find the optima of $\tilde{F}_{+1}(x) = \sum x_i^2 \log x_i^2$, which are known from standard information theory to be 0 and $n \log n$.

B Proof of Lemma 5.2

Proof. Define the function $g : \mathbb{R} \mapsto \mathbb{R}$ as

$$g(z) = -\frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{n}} + z \right) \log \left| \frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{n}} + z \right) \right| .$$

Let $g'(z)$ define the derivative of g (where exists, namely for $z \neq -1/\sqrt{n}$). Clearly,

$$g'(z) = -\frac{1}{\sqrt{n}} \left(\log \left| \frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{n}} + z \right) \right| + 1 \right) .$$

We split the sum in (5.3) to $i \in I^+ := \{i' : \varepsilon_{i'} \geq 0\}$ and $i \in I^- := [n] \setminus I^+$. For the former case, note that g is monotonically increasing in the range $[0, 1/2]$. Hence,

$$\sum_{i \in I^+} g(\varepsilon_i) \geq |I^+| \cdot g(0) = \frac{|I^+|}{n} \log n . \quad (\text{B.1})$$

(We also used the fact that $|\varepsilon_i| < 1/2$ for all i .) For $i \in I^-$, define $z_0 < -1/\sqrt{n}$ to be the unique number such that

$$\frac{g(0) - g(z_0)}{-z_0} = g'(z_0) . \quad (\text{B.2})$$

(This is the unique point to the left of $-1/\sqrt{n}$ such that the tangent line to g at that point intersects the vertical line $z = 0$ at $(0, g(0)) = (0, \frac{1}{n} \log n)$.) It is trivial to show that

$$-1 \leq z_0 \leq -\frac{C_1}{\sqrt{n}} \log n \quad (\text{B.3})$$

for some global $C_1 > 0$. It is also trivial to show that for all $z < 0$,

$$g(z) \geq -g(z_0) + (z - z_0)g'(z_0) = g(0) + z \cdot g'(z_0) = \frac{1}{n} \log n - \frac{z}{\sqrt{n}} \left(\log \left| \frac{1}{\sqrt{n}} \left(\frac{1}{\sqrt{n}} + z_0 \right) \right| + 1 \right) ,$$

namely, the graph of g (in the left half plane) lies above the tangent at z_0 . This implies, also using (B.3), that

$$\sum_{i \in I^-} g(\varepsilon_i) \geq \frac{|I^-|}{n} \log n + \sum_{i \in I^-} \frac{\varepsilon_i}{\sqrt{n}} (C_2 \log n + C_3) ,$$

for some global $C_2 > 0$ and C_3 . By setting C_0 appropriately and recalling that $\|\varepsilon\|_1 \leq \sqrt{n} \|\varepsilon\|_2$, we get

$$\sum_{i \in I^-} g(\varepsilon_i) \geq \frac{|I^-|}{n} \log n - \frac{1}{4} \log n . \quad (\text{B.4})$$

Combining (B.1) and (B.4), we conclude the required. \square